

# Unsupervised Single Image Deraining with Self-supervised Constraints

Xin Jin, Zhibo Chen, Jianxin Lin, Zhikai Chen, Wei Zhou  
University of Science and Technology of China

jinxustc@mail.ustc.edu.cn, chenzhibo@ustc.edu.cn, {linjx, czk654, weichou}@mail.ustc.edu.cn

## Abstract

Most existing single image deraining methods require learning supervised models from a large set of paired synthetic training data, which limits their generality, scalability and practicality in real-world multimedia applications. Besides, due to lack of labeled-supervised constraints, directly applying existing unsupervised frameworks to the image deraining task will suffer from low-quality recovery. Therefore, we propose an Unsupervised Deraining Generative Adversarial Network (UD-GAN) to tackle above problems by introducing self-supervised constraints from the intrinsic statistics of unpaired rainy and clean images. Specifically, we firstly design two collaboratively optimized modules, namely Rain Guidance Module (RGM) and Background Guidance Module (BGM), to take full advantage of rainy image characteristics: The RGM is designed to discriminate real rainy images from fake rainy images which are created based on outputs of the generator with BGM. Simultaneously, the BGM exploits a hierarchical Gaussian-Blur gradient error to ensure background consistency between rainy input and de-rained output. Secondly, a novel luminance-adjusting adversarial loss is integrated into the clean image discriminator considering the built-in luminance difference between real clean images and de-rained images. Comprehensive experiment results on various benchmarking datasets and different training settings show that UD-GAN outperforms existing image deraining methods in both quantitative and qualitative comparisons.

## 1. Introduction

Single image deraining is important for many outdoor multimedia applications such as surveillance, pedestrian detection and autonomous driving, etc. Recently, many deep learning-based deraining methods have been proposed to address this problem [16, 7, 8, 44, 50, 27, 6, 26]. These methods are mainly trained on synthetic rainy-clean image pairs (Figure 1 (a)) in a supervised manner and then applied in real-world rainy scenarios, which causes several limitations: (1) The manually synthetic rain shapes usually differ



Figure 1. (a) Synthetic rainy image and corresponding clean image. (b) Real-world rainy image and random clean image.

from the real ones in nature due to the rain distribution gap between them, which causes these fully-supervised deraining approaches to have limited ability to remove unknown rain from real-world rainy images. (2) The pattern and style of rain are various, which makes it difficult to treat all scenarios with just one fully-supervised deraining model. For example, the model trained on the heavy rain could not be directly applied in the light-rainy scenario, and vice versa. Therefore, we attempt to handle with the deraining task from a completely different perspective of resorting to unsupervised learning with unpaired real-world data (Figure 1 (b)), so that the problems mentioned above could be well solved.

Unfortunately, although significant success has been achieved in unsupervised learning-based image processing models, such as CycleGAN [53] and WESPE [21], they still fail to surpass previous supervised models in deraining task. There exist two major reasons: (1) Unsupervised training schemes usually suffer from the under-constrained problem since supervised constraints such as mean-square error (MSE) between the output and the ground truth cannot be applied directly, which often results in unwanted artifacts as shown in Figure 2 (a). (2) Most unsupervised networks are designed to learn a one-to-one transformation, such as horse-to-zebra, day-to-night, etc. But for deraining, they hardly capture the varied transformations from rainy to clean because the direction, shape and density of rain streaks are various as shown in Figure 2 (b).

To address these two problems, in this paper, we propose a novel perspective to achieve unsupervised deraining: *Instead of relying solely on pure unsupervised domain transformations, we introduce self-supervised constraints from the intrinsic statistics of unpaired rainy and clean*

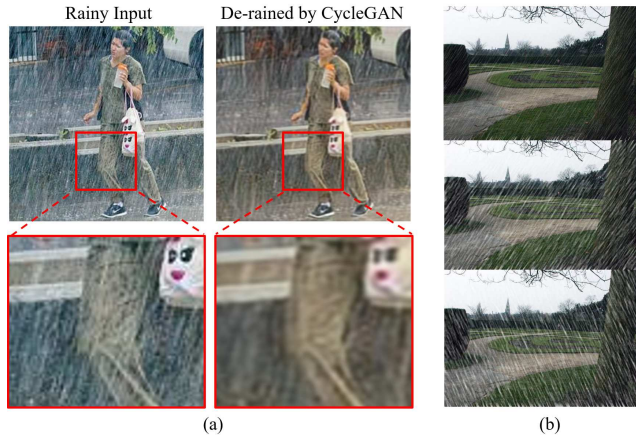


Figure 2. (a) Directly using CycleGAN [53] to do image deraining will leave lots of residual rain streaks and generate unacceptable artifacts. (b) The direction, shape and density of rain streaks are various.

images to guide deraining, which solves the first problem of under-constraint. Specifically, we propose an end-to-end learning Unsupervised Deraining Generative Adversarial Network (UD-GAN) with two collaboratively optimized modules: Rain Guidance Module (RGM) and Background Guidance Module (BGM). RGM indirectly constrains the solution space of generated de-rained images by constraining the difference (i.e. removed rain streaks) between rainy input and de-rained output, which further solves the second negative influence on unsupervised deraining due to the variety of rain streaks. BGM ensures background consistency by imposing a hierarchical Gaussian-Blur gradient error between rainy input and de-rained output. Considering the built-in luminance difference between real clean images and de-rained images, a luminance-adjusting adversarial loss is designed for obtaining more natural and realistic de-rained results. The contributions of this paper are summarized as follows:

- To the best of our knowledge, this study is the first data-driven attempt to unsupervised learning for deraining task trained with unpaired image sets.
- By learning from the intrinsic statistics of raw data in a self-supervised manner, we provide a novel perspective for unsupervised training, which opens a new way to single image deraining, bringing it closer to practical applications.
- Extensive performance evaluation on both synthetic and real-world datasets validates the effectiveness of our method. Especially for the improvement of subjective effects in real-world rainy scenes, UD-GAN greatly exceeds existing supervised methods and can be easily generalized to other computer vision tasks.

## 2. Related Work

### 2.1. Single Image Deraining

**Traditional Methods:** Traditional prior-based methods have been proposed in the literature to deal with single image deraining problem, such as sparse coding-based methods [18, 30, 54], low-rank representation-based methods [4, 49] and gaussian mixture model-based (GMM) methods [28, 29], etc. The main limitation of existing prior-based methods is that they often tend to have under de-rained effect by leaving residual rain streaks [24] or over-smooth image details [30].

**Deep Neural Network (DNN):** The renaissance of DNN remarkably accelerated the progress of deraining task: Fu et al. [7] proposed a learning-based rain removal solution, then they also combined ResNet [16] and focused on high-frequency details while deraining in DetailsNet [8]. [44] proposed a deep recurrent network named JORDER to remove rain streaks progressively. This year, [50] presented a density-aware multi-stream connected network called DIDMDN for joint rain density estimation and deraining. [27] proposed a recurrent neural network with dilated convolution [47] and Squeeze-and-Excitation (SE) blocks [17], called RESCAN. [6] designed a residual-guide network (RGN) to achieve a coarse-to-fine deraining. [26] introduced a non-locally enhanced encoder-decoder network (NLEDN) for more accurate rain removal. In general, the common idea of above methods is limited to view deraining as a regression problem and learn a mapping between synthetic rainy inputs and ground truths using a CNN structure in a fully-supervised manner, which limits their generality, scalability and practicality in real-world rainy scenes.

### 2.2. Unsupervised Learning

Recently, unsupervised learning-based image processing applications emerged with promising performance [51, 38, 48, 31, 34, 12, 45, 43]. These methods usually can be divided into two categories: One is to utilize unsupervised learning to estimate the data distribution for data enhancement, and then use the enhanced data to train the model [3], which has limited scalability because the distribution of some data is difficult to be estimated such as rain. Another is to use unpaired data to achieve domain transfer, DualGAN [46] and CycleGAN [53] are two classic GAN-based [11] works belonging to this category, and both of them use a pair of GANs to learn the transformation. However, training GANs is highly unstable [36] and thus using two GANs simultaneously escalates in instability. Moreover, it is impossible to directly apply DualGAN or CycleGAN for image deraining, because of the above-mentioned defects existing in unsupervised learning: (1) under-constrained problem and (2) hard to capture the varied transformations from rainy inputs to clean outputs due to rain streaks' diversity.

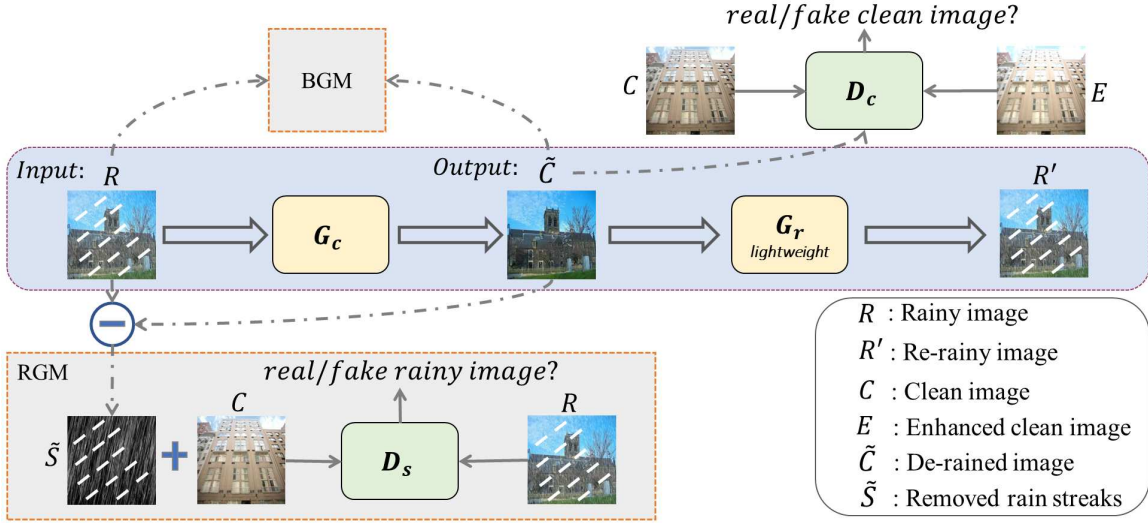


Figure 3. An overview of the proposed UD-GAN with two collaboratively optimized Rain Guidance Module (RGM) and Background Guidance Module (BGM). The RGM indirectly helps the generator  $G_c$  de-rain the rainy input by using discriminator  $D_s$  to constrain the removed rain streaks, while the BGM ensures the background consistency between rainy input and de-rained output.  $D_c$  is a clean image discriminator with luminance adjustment function. The lightweight rain-added generator  $G_r$  is to avoid style/color variation.

### 2.3. Self-supervised Learning

The debilitating limitation of supervised learning and the defect of unsupervised learning together necessitate the need for self-supervised learning, which is a form of unsupervised learning where the data provides the supervision. Self-supervised learning has demonstrated success in many computer vision applications [1, 5, 32, 39, 37, 10, 9]. Among them, [37] introduced a new approach to supervising neural networks without any labeled examples by specifying constraints that are derived from prior domain knowledge, e.g., from known laws of physics. [10] put forward an idea of performing self-supervised learning of visual features by mining a large scale corpus of multi-modal (i.e. text and image) documents. [9] leveraged semantic feature in self-supervised manner to achieve the recognition of 2D image rotation.

Inspired by these works, in this paper, we try to fully exploit the intrinsic characteristics of original data in a self-supervised manner, which achieves the goal of transforming rainy inputs to clean outputs without any paired data.

## 3. Unsupervised Deraining Generative Adversarial Network (UD-GAN)

As illustrated in Figure 3, our goal is learning to transform images from the rainy domains  $R$  to the target clean domain  $C$  given random and unpaired training samples  $\{r_i\}_{i=1}^N \in R$  and  $\{c_j\}_{j=1}^M \in C$ . The first generator  $G_c$  is used to transform rainy inputs to the clean outputs, which captures a deraining transformation:  $R \rightarrow C$ . Correspondingly, an adversarial discriminator  $D_c$  is used to distinguish

between real clean images  $C$  and fake de-rained images  $\tilde{C}$ , where  $\{\tilde{c}\} \in \tilde{C}$  represents a fake de-rained result:

$$\tilde{c} = G_c(r), \quad D_c(c, \tilde{c}) = \text{real/fake}. \quad (1)$$

However, training deraining generator  $G_c$  with adversarial cost alone may introduce visual artifacts [22] in certain regions of the generated de-rained output, but the clean image discriminator  $D_c$  can still end up classifying it as real data rather than generated data, which is unacceptable. To solve this problem, self-supervised constraints from the intrinsic statistics of deraining problem are introduced in the following sections.

### 3.1. Self-supervision by Rainy Image

#### 3.1.1 Rain Guidance Module (RGM)

As discussed above, existing unsupervised frameworks tend to suffer from the under-constrained problem due to lack of labeled-supervised constraints and hard to capture the varied transformations from rainy inputs to clean outputs due to the diversity of rain streaks. Hence, we introduce a Rain Guidance Module (RGM) to take full advantage of the intrinsic statistics of original rainy inputs, and in turn use the learned rain characteristics to guide in better deraining.

In detail, based on the widely used rain model [19, 30, 28, 54, 23, 50, 41]:  $R = C + S$ , we leverage the inner interdependency between clean images  $C$  and rain streaks  $S$  to indirectly constrain the de-rained outputs  $\tilde{C}$  of the deraining generator  $G_c$ . Refer to Figure 3, RGM mainly depends on an independent discriminator  $D_s$ . We first obtain the difference (i.e. removed rain streaks  $\tilde{S}$ ) between rainy inputs  $R$

and de-rained outputs  $\tilde{C}$  by  $\tilde{S} = R - \tilde{C}$ . Then we superimpose rain streaks  $\tilde{S}$  on the real clean images  $C$  to create fake rainy images  $\tilde{S} + C$ , and distinguish them from real rainy images  $R$  by rain streaks discriminator  $D_s$ :

$$\tilde{s} = r - \tilde{c}, \quad D_s(r, \tilde{s} + c) = \text{real/fake}, \quad (2)$$

where  $\{\tilde{s}\} \in \tilde{S}$ ,  $\{\tilde{c}\} \in \tilde{C}$ ,  $\{r\} \in R$  and  $\{c\} \in C$ . We hold the view that if the deraining generator  $G_c$  is trained well enough, when we superimpose the removed rain streaks  $\tilde{S}$  on any real clean image  $C$ , the obtained fake rainy images  $\tilde{S} + C$  should be indistinguishable from the real rainy images  $R$ . That is, RGM authentically acts as a "supervising teacher" to indirectly help the deraining generator  $G_c$  obtain better de-rained outputs  $\tilde{C}$  by encouraging the removed rain streaks  $\tilde{S} = R - \tilde{C}$  to be close to the real rain streaks.

Correspondingly, a rain guidance loss  $\mathcal{L}_{guid}^R$  is defined as follows, which optimizes the deraining generator  $G_c$  and rain streaks discriminator  $D_s$  simultaneously:

$$\begin{aligned} \mathcal{L}_{guid}^R(G_c, D_s) = & \mathbb{E}_{r \sim p_{data}(r)} [\log D_s(r)] \\ & + \mathbb{E}_{r, c \sim p_{data}(r, c)} [\log(1 - D_s((r - G_c(r) + c)))] \end{aligned} \quad (3)$$

This guidance from RGM essentially can be viewed as a kind of adversarial collaboration: the more realistic the removed rain streaks  $\tilde{S}$  are, the cleaner the de-rained results  $\tilde{C}$  are, and vice versa.

### 3.1.2 Background Guidance Module (BGM)

In addition to the correct transformation from rainy to clean domain, another important goal in image deraining is to ensure content consistency and avoid losing important details after deraining. To achieve this goal, we design another Background Guidance Module (BGM) to further utilize the input rainy images to provide more reasonable and reliable self-supervised constraints for the deraining generator  $G_c$ .

Inspired by [24, 40, 7, 8], we found that after applying an appropriate low-pass filter such as Gaussian blur kernel<sup>1</sup> or Guided Filtering [14], low-pass versions of both the rainy image and the clean image are rain-free and approximately equal, they only contain the background/content features. We can take advantage of this fact to guide the deraining process. Specifically, we hierarchically use the Gaussian blur kernels with different scales  $\sigma$  to filter rainy input  $R$  and de-rained output  $\tilde{C}$ , obtaining their background features respectively. Figure 4 shows that as we increase in Gaussian blur scale  $\sigma$ , not only the background features start to look alike, but also the average gradient error between them decreases. Based on this, we naturally form a guidance for the deraining generator  $G_c$  through a background guidance loss  $\mathcal{L}_{guid}^B$ , which enforces the Gaussian-Blur gradients of rainy input and de-rained output to match at different scales  $\sigma$ :

<sup>1</sup><http://homepages.inf.ed.ac.uk/rbf/HIPR2/log.htm>

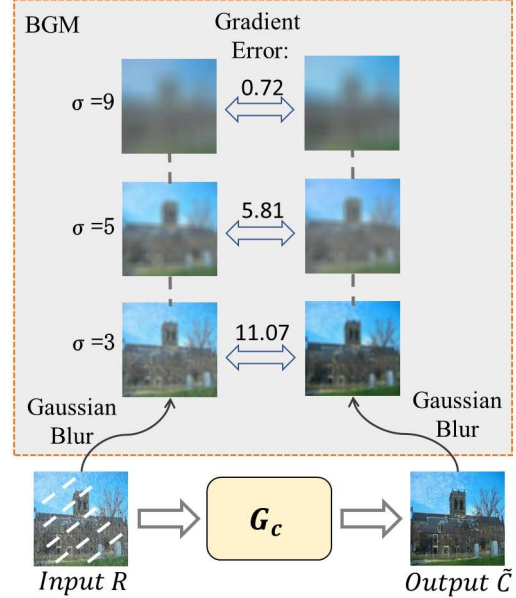


Figure 4. Background Guidance Module (BGM), which hierarchically uses the gradient errors at different Gaussian-Blur levels to ensure the content consistency between rainy inputs  $R$  and de-rained outputs  $\tilde{C}$ .

$$\mathcal{L}_{guid}^B(G_c) = \sum_{\sigma=3,5,9} \lambda_{\sigma} |\nabla B_{\sigma}(r) - \nabla B_{\sigma}(G_c(r))|, \quad (4)$$

where  $\nabla, B_{\sigma}$  denote the gradient computation and Gaussian blur operation.  $\lambda_{\sigma}$  values are used to balance errors at different Gaussian-Blur levels. We believe that it is necessary to utilize these gradient errors hierarchically, because different levels of background features contain different levels of important detail information. Hence, based on experimental attempts, we set  $\lambda_{\sigma}$  as [0.01, 0.1, 1] for  $\sigma = 3, 5, 9$ , respectively.

### 3.2. Self-supervision by Clean Image

As shown in Eq. 1, the deraining generator  $G_c$  and the clean image discriminator  $D_c$  form a complete GAN [11], an adversarial loss should be applied to them, its value indicates what extent the de-rained output of deraining generator  $G_c$  looks like a clean image. However, we observe that simply training  $D_c$  through a standard adversarial loss like in [11, 53, 22] to separate generated de-rained images and true clean images is not sufficient due to the built-in luminance difference between them. In detail, we find that (1) the real rainy images are almost cloudy, their average luminance is usually lower. If we directly use the real clean images (usually sunny in our collected real-world dataset) to constraint the de-rained outputs through a standard adversarial loss, the de-rained outputs are often too bright and do not match the cloudy day (Figure 5 (left)), (2) the rain



Figure 5. Left: directly using the real clean images (usually sunny in dataset) to constraint the de-rained outputs often causes over-bright results (b) and don’t match the cloudy situation (c). Right: by enhancing luminance in clean images  $\{c_j\}_{j=1}^M \in C$ , we additionally generate a negative training sample set  $E$ :  $\{e_k\}_{k=1}^K \in E$ .

streaks of the synthetic rainy image always appear as some bright lines (Figure 1 (a)). So, those de-rained images with higher illuminance may leave more residual rain streaks.

We explore to leverage some ”clean image constraints” to circumvent the above problems. Specifically, from the clean training samples  $\{c_j\}_{j=1}^M \in C$ , we additionally generate a negative training sample set  $E$ :  $\{e_k\}_{k=1}^K \in E$  by enhancing the luminance [13] of the images in  $C$ . As shown in Figure 5 (right), during training, the clean image discriminator  $D_c$  should maximize the probability of assigning the correct label to fake de-rained images  $\tilde{C}$ , real clean images  $C$  and luminance-enhanced clean images  $E$ , such that the deraining generator  $G_c$  can be guided correctly in transforming the rainy input to the clean output. Therefore, we re-define a luminance-adjusting adversarial loss as follows:

$$\begin{aligned} \mathcal{L}_{lum.adv}(G_c, D_c) = & \mathbb{E}_{c \sim p_{data}(c)} [\log D_c(c)] \\ & + \mathbb{E}_{e \sim p_{data}(e)} [\log(1 - D_c(e))] \\ & + \mathbb{E}_{r \sim p_{data}(r)} [\log(1 - D_c(G_c(r)))]. \end{aligned} \quad (5)$$

We discuss that adding such a luminance-adjusting adversarial loss ensures that (1) the de-rained results in real-world rainy scene have more realistic and natural luminance, (2) the de-rained results for synthetic rainy images leave less residual rain streaks. Figure 7,8 in the experiment section validate the improvement of subjective effects.

### 3.3. Loss Function

Except for the above-mentioned rain guidance loss  $\mathcal{L}_{guid}^R$ , background guidance loss  $\mathcal{L}_{guid}^B$  and luminance-adjusting adversarial loss  $\mathcal{L}_{lum.adv}$ , the final loss function for UD-GAN also contains a cycle consistency loss  $\mathcal{L}_{cyc}$  to cope with substantial style/color variations between rainy input and de-rained output.

Inspired by [53, 21], we employ another generator  $G_r$  to learn the inverse rain-added transformation:  $r' = G_r(\tilde{c})$ , where  $\{r'\} \in R'$  representing the reconstructed re-rainy results as shown in Figure 3. Considering training complexity and time complexity, the rain-added generator  $G_r$  is designed as a lightweight network compared to the deraining generator  $G_c$  and doesn’t have corresponding discriminator. We optimize  $G_r$  only through a consistency loss:

$$\mathcal{L}_{cyc}(G_c, G_r) = \mathbb{E}_{r \sim p_{data}(r)} [\|G_r(G_c(r)) - r\|_1]. \quad (6)$$

In summary, the final loss for UD-GAN is a weighted sum of the above four losses, where  $w_1 \sim w_4$  are the weights to balance different losses. Based on experimental attempts, we finally set  $w_1 \sim w_4$  as 1, 5, 1, 0.5:

$$\mathcal{L}_{total} = w_1 \mathcal{L}_{guid}^R + w_2 \mathcal{L}_{guid}^B + w_3 \mathcal{L}_{lum.adv} + w_4 \mathcal{L}_{cyc}. \quad (7)$$

## 3.4. Network Architecture and Training Details

Source codes are coming soon. During training, the weights of our model are all initialized using the technique described in [15]. We adopt an Adam [25] solver to minimize the whole objective function in Eq. 7. We set the mini-batch size to 1, the size of the input image is 512\*512. The entire network is trained on two Nvidia 1080Ti GPUs based on Pytorch framework. The initial learning rate  $\alpha$  is 0.0002 and decreases as the number of iterations increases. We train the model over 200,000 iterations, until it well converges.

## 4. Experiment

### 4.1. Dataset and Evaluation Metrics

We carry out deraining experiments below on four widely used synthetic datasets and a real-world dataset. Rain800 [50, 27], Rain12, Rain100L and Rain100H [8, 44] are the synthetic datasets with various synthetic rain streaks. For real-world dataset, we collect 784 real-world rainy images (including some snowing images) from the Internet and the previous studies [8, 44, 27], which are diverse in content and rain. All datasets are divided into training and testing set with a ratio of 7:1. To highlight the generalization ability of our model in both synthetic and real-world rainy scenarios, we introduce three training schemes in total: (1) Training only on synthetic datasets, UD-GAN<sub>syn</sub>, (2) Training only on real-world datasets, UD-GAN<sub>real</sub>, (3) Training on all datasets, UD-GAN. All the schemes are evaluated on both synthetic and real-world test datasets.

Deraining performance on the synthetic data is evaluated in terms of PSNR [20] and SSIM [42]. Performance on real-world images is evaluated visually since the ground truth images are not available. We compare UD-GAN with the following state-of-the-art methods in the same test environment: image decomposition (ID) [24] (TIP’12), discriminative sparse coding (DSC) [30] (ICCV’15), gaussian mixture model (GMM) [28] (CVPR’16), CNN method (CNN) [7] (TIP’17), DetailsNet [8] (CVPR’17), JORDER [44] (CVPR’17), DID-MDN [50] (CVPR’18), RESCAN [27] (ECCV’18), RGN [6] (ACMMM’18) and NLEDN [26] (ACMMM’18).

Table 1. Quantitative comparison with existing methods on Rain800, Rain12, Rain100L and Rain100H. The three best performing methods are marked in red, blue, and green, respectively.

Dataset	Rain800		Rain12		Rain100L		Rain100H	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Metric	18.88	0.5832	27.21	0.7548	23.13	0.7023	14.02	0.5219
ID [24]	18.56	0.5996	30.02	0.8745	24.16	0.8721	15.66	0.5467
DSC [30]	20.46	0.7297	32.02	0.9155	29.11	0.8809	14.26	0.4225
CNN [7]	19.22	0.6418	31.19	0.8917	28.70	0.8914	16.08	0.6117
DetailsNet [8]	21.16	0.7320	33.75	0.9319	34.85	0.9508	22.82	0.7409
JORDER [44]	22.29	0.7922	36.02	0.9617	36.11	0.9707	23.45	0.7507
DID-MDN [50]	25.12	0.8845	36.14	0.9634	36.14	0.9711	26.69	0.8774
RESCAN [27]	24.09	0.8410	35.87	0.9522	36.12	0.9639	26.43	0.8458
RGN [6]	24.04	0.8812	29.45	0.9380	33.16	0.9631	25.25	0.8418
NLEDN [26]	24.09	0.8766	33.16	0.9192	36.57	0.9747	27.03	0.8819
UD-GAN <sub>syn</sub>	25.02	0.8797	36.35	0.9537	36.20	0.9723	26.81	0.8838
UD-GAN <sub>real</sub>	24.78	0.8817	36.21	0.9481	35.95	0.9683	26.61	0.8860
UD-GAN	25.98	0.9093	37.13	0.9644	37.28	0.9753	27.75	0.8931



Figure 6. UD-GAN<sub>real</sub> that trained only on real rainy data shows a little bit lower PSNR (around 0.4dB) compared with other supervised methods on synthetic datasets, but achieves much better subjective results for real-world rainy images. As shown above, the details of the boy’s face and the car’s logo have been clearly preserved in result (c), compared with DID-MDN’s blur result (b).

## 4.2. Comparison Results

**Synthetic Data:** Table 1 shows the quantitative results of different methods on synthetic datasets Rain800, Rain12, Rain100L and Rain100H. We can observe that our method is able to perform equally well when only training on the synthetic datasets (UD-GAN<sub>syn</sub>) or real-world datasets (UD-GAN<sub>real</sub>) compared to other fully-supervised methods, and even outperform them (over 0.7dB PSNR gain) when training on both synthetic and real-world datasets. That is because (1) the involvement of real-world rainy images helps our model break away from the limitation of synthetic data, reaching the effect of data augmentation, (2) based on the reliable intrinsic statistics of unpaired rainy and clean images, we provide more generalized, appropriate and reasonable self-supervised constraints for the network than existing fully-supervised methods.

It can be noted that, although the PSNR and SSIM of UD-GAN<sub>real</sub> on some synthetic datasets is lower than that of the best STOA supervised solution DID-MDN [50], but the subjective de-rained effect of UD-GAN<sub>real</sub> is obviously better than DID-MDN as shown in Figure 6.

In Figure 7 (top three rows), we select most advanced methods and most difficult synthetic rainy images to further

Table 2. Objective results of removing different components of UD-GAN.

Methods	PSNR				
	Ours - {RGM,BGM,lum}	Ours - RGM	Ours - BGM	Ours - lum	Ours
Rain800	23.58	24.85	25.07	25.77	25.98
Rain12	34.61	35.13	35.31	36.81	37.13
Rain100L	34.67	35.89	35.86	36.72	37.28
Rain100H	23.72	25.36	25.62	27.14	27.75
SSIM					
Rain800	0.8133	0.8749	0.8792	0.8926	0.9093
Rain12	0.9421	0.9553	0.9501	0.9591	0.9644
Rain100L	0.9379	0.9525	0.9626	0.9692	0.9753
Rain100H	0.7249	0.8359	0.8323	0.8697	0.8931

show that UD-GAN promises the most satisfactory subjective de-rained effect, which effectively removes rain streaks while preserving better details.

**Real-world Data:** To test the practicability of UD-GAN, we also evaluate its performance on real-world rainy images. Figure 7 (bottom five rows) shows some de-rained results on the real-world test dataset: DetailsNet [8] tends to leave residual rain streaks in the background. DID-MDN [50] over-smooths some important details such as building structures as shown in the last row, and cannot handle these snow-like raindrops as shown in the first, second rows. JORDER [44] and RESCAN [27] suffer from unexpected artifacts on the de-rained results as shown in the middle and last row (please zooming-in to observe). In contrast, UD-GAN effectively restores clean background with rich texture details while promising more natural and realistic luminance, which significantly improves the subjective effects and greatly surpasses other methods in terms of clarity and visibility.

## 4.3. Ablation Study

To verify the effectiveness of the proposed Rain Guidance Module (RGM), Background Guidance Module (BGM) and luminance-adjusting adversarial loss in UD-GAN, we implement four ablated schemes for comparison. Due to space limitation, here we abbreviate the complete UD-GAN as “Ours”:



Figure 7. Deraining results of different methods on rainy images from the synthetic datasets (top three rows), and real-world dataset (bottom five rows). PSNR/SSIM have been calculated and attached below synthetic samples.

- **Ours - {RGM,BGM,lum}**. Benchmark scheme without RGM, BGM or luminance-adjusting loss.
- **Ours - RGM**. Remove Rain Guidance Module.
- **Ours - BGM**. Remove Background Guidance Module.
- **Ours - lum**. Remove luminance-adjusting component in the adversarial loss.

Table 2 shows that the complete UD-GAN achieves 2.40/2.52/2.61/4.03dB, 1.13/2.00/1.39/2.39dB, 0.91/1.82/1.42/2.13dB and 0.21/0.32/0.56/0.61dB PSNR gain over four ablated baselines (Ours - {RGM,BGM,lum},

Ours - RGM, Ours - BGM and Ours - lum) on Rain800, Rain12, Rain100L and Rain100H respectively.

Subjective comparisons are presented in Figure 8: Benchmark scheme produces unacceptable artifacts. RGM makes the restored background more clear and visible. BGM promises more realistic and relatively richer textural content. Luminance-adjusting adversarial loss helps the de-rained output has natural luminance and looks cleaner.

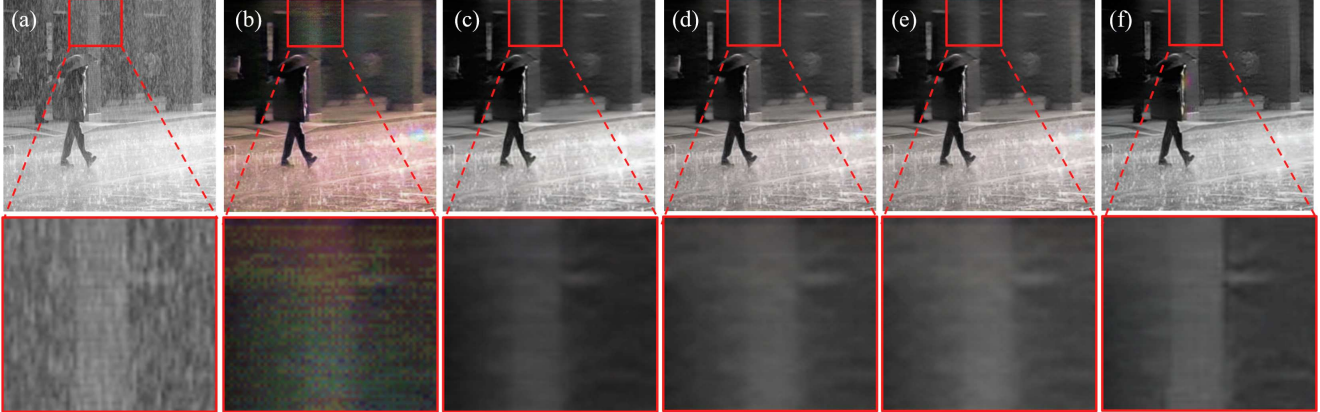


Figure 8. Subjective results of removing different components of UD-GAN. From left to right: (a) rainy input, (b) Ours - {RGM, BGM, lum}, (c) Ours - RGM, (d) Ours - BGM, (e) Ours - lum, and (f) the complete UD-GAN.

#### 4.4. User Study

For a more comprehensive qualitative evaluation, we conduct three user studies to respectively demonstrate the effectiveness of  $UD\text{-GAN}_{syn}$ ,  $UD\text{-GAN}_{real}$  and  $UD\text{-GAN}$  in generating visually attractive results. To build the subjective database, we choose 12 rainy images randomly from synthetic and real-world test data with a ratio of 1:1, and each image is de-rained by DetailsNet [8], JORDER [44], DID-MDN [50], RESCAN [27],  $UD\text{-GAN}_{syn}$ ,  $UD\text{-GAN}_{real}$  and  $UD\text{-GAN}$  separately. Then we respectively compare the de-rained results of  $UD\text{-GAN}_{syn}$ ,  $UD\text{-GAN}_{real}$  and  $UD\text{-GAN}$  with other 4 methods in three independent user studies (No.1, No.2, No.3). Refer to the subjective experiment design [27] and assessment criteria [2, 33] in the previous studies, we show the original rainy image together with its five de-rained results using different methods to 10 non-expert subjects, then they are instructed to vote for the best de-rained result with the least rain streaks and the clearest texture details. As shown in Table 3,  $UD\text{-GAN}_{syn}$ ,  $UD\text{-GAN}_{real}$  and  $UD\text{-GAN}$  respectively gets the most 88, 93, 101 votes in three independent user studies, which demonstrates the superiority of our method in synthesizing subjective high-quality de-rained images.

#### 4.5. Time Complexity Comparisons

Computational time comparisons are shown in Table 4. The proposed UD-GAN is comparable to other methods because only  $G_c$  works when testing, it only takes about 0.25s on average to process a rainy image of size 512\*512.

#### 4.6. Extension

We validate that UD-GAN can generalize to other low-level image processing tasks such as **image denoising**, and also can be used to pre-process for high-level vision such as **action recognition**. Experimental results can be found in Table 5.

Table 3. Results of three user studies (Voting number of different methods, the higher the better).

	DetailsNet	JORDER	DID-MDN	RESCAN	UD-GAN <sub>syn</sub>	UD-GAN <sub>real</sub>	UD-GAN
No.1	0	2	24	6	88	-	-
No.1	2	1	23	1	-	93	-
No.3	1	1	14	3	-	-	101

Table 4. Computational time for different methods averaged on 200 images with size 512\*512.

	DetailsNet	JORDER	DID-MDN	RESCAN	UD-GAN
512*512 (GPU)	0.34s	1.88s	0.28s	4.75s	<b>0.25s</b>

Table 5. Denoising results (SSIM) on BSD68 [35].

BSD68 [35]	$\sigma=15$	$\sigma=25$	$\sigma=50$
DnCNN [52]	0.8826	0.8190	0.7076
UD-GAN	0.8822	0.8198	0.7088

## 5. Conclusion

In this paper, we tackle the single image deraining problem in an unsupervised manner with an end-to-end learned model, i.e. Unsupervised Deraining GAN (UD-GAN). Compared to existing supervised approaches which attempt to learn a mapping between synthetic rainy inputs and corresponding ground truths, we provide the network with more reliable and reasonable self-supervised constraints from the intrinsic statistics of original data through two collaboratively optimized modules BGM & RGM and a luminance-adjusting adversarial loss. Sufficient experiments and comparisons are performed on both synthetic and real-world datasets to demonstrate the effectiveness, generalization and practicability of the proposed UD-GAN. In future, we plan to extend the proposed self-supervision algorithm to a wider range of unsupervised image restoration tasks including deblurring, dehazing and super-resolution.



## References

- [1] P. Agrawal, J. Carreira, and J. Malik. Learning to see by moving. In *ICCV*, pages 37–45, 2015. 3
- [2] R. I.-R. BT. Methodology for the subjective assessment of the quality of television pictures. 2002. 8
- [3] J. Chen, J. Chen, H. Chao, and M. Yang. Image blind denoising with generative adversarial network based noise modeling. In *CVPR*, pages 3155–3164, 2018. 2
- [4] Y.-L. Chen and C.-T. Hsu. A generalized low-rank appearance model for spatio-temporally correlated rain streaks. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1968–1975. IEEE, 2013. 2
- [5] C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, pages 1422–1430, 2015. 3
- [6] Z. Fan, H. Wu, X. Fu, Y. Huang, and X. Ding. Residual-guide network for single image deraining. In *2018 ACM Multimedia Conference on Multimedia Conference*, pages 1751–1759. ACM, 2018. 1, 2, 6
- [7] X. Fu, J. Huang, X. Ding, Y. Liao, and J. Paisley. Clearing the skies: A deep network architecture for single-image rain removal. *IEEE Transactions on Image Processing (TIP)*, 26(6):2944–2956, 2017. 1, 2, 4, 6
- [8] X. Fu, J. Huang, D. Zeng, Y. Huang, X. Ding, and J. Paisley. Removing rain from single images via a deep detail network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 2, 4, 5, 6, 8
- [9] S. Gidaris, P. Singh, and N. Komodakis. Unsupervised representation learning by predicting image rotations. *ICLR*, 2018. 3
- [10] L. Gomez, Y. Patel, M. Rusiñol, D. Karatzas, and C. Jawahar. Self-supervised learning of visual features through embedding images into text topic spaces. *CVPR*, 2017. 3
- [11] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems (NIPS)*, pages 2672–2680, 2014. 2, 4
- [12] V. C. Guizilini and F. T. Ramos. Unsupervised feature learning for 3d scene reconstruction with occupancy maps. In *AAAI*, pages 3827–3833, 2017. 2
- [13] X. Guo, Y. Li, and H. Ling. Lime: Low-light image enhancement via illumination map estimation. *IEEE Transactions on Image Processing*, 26(2):982–993, 2017. 5
- [14] K. He, J. Sun, and X. Tang. Guided image filtering. *IEEE transactions on pattern analysis & machine intelligence*, (6):1397–1409, 2013. 4
- [15] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision (ICCV)*, pages 1026–1034, 2015. 5
- [16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 1, 2
- [17] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. *CVPR*, 2017. 2
- [18] D.-A. Huang, L.-W. Kang, Y.-C. F. Wang, and C.-W. Lin. Self-learning based image decomposition with applications to single image denoising. *IEEE Transactions on multimedia*, 16(1):83–93, 2014. 2
- [19] D.-A. Huang, L.-W. Kang, M.-C. Yang, C.-W. Lin, and Y.-C. F. Wang. Context-aware single image rain removal. In *Multimedia and Expo (ICME), 2012 IEEE International Conference on*, pages 164–169. IEEE, 2012. 3
- [20] Q. Huynh-Thu and M. Ghanbari. Scope of validity of psnr in image/video quality assessment. *Electronics letters*, 44(13):800–801, 2008. 5
- [21] A. Ignatov, N. Kobyshev, R. Timofte, K. Vanhoey, and L. Van Gool. Wespe: weakly supervised photo enhancer for digital cameras. *CVPR*, 2018. 1, 5
- [22] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *CVPR*, 2017. 3, 4
- [23] X. Jin, Z. Chen, J. Lin, J. Chen, W. Zhou, and C. Shan. A decomposed dual-cross generative adversarial network for image rain removal. *The British Machine Vision Conference (BMVC)*, 2018. 3
- [24] L.-W. Kang, C.-W. Lin, and Y.-H. Fu. Automatic single-image-based rain streaks removal via image decomposition. *IEEE Transactions on Image Processing*, 21(4):1742–1755, 2012. 2, 4, 5, 6
- [25] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *ICLR*, 2015. 5
- [26] G. Li, X. He, W. Zhang, H. Chang, L. Dong, and L. Lin. Non-locally enhanced encoder-decoder network for single image de-raining. In *2018 ACM Multimedia Conference on Multimedia Conference*, pages 1056–1064. ACM, 2018. 1, 2, 6
- [27] X. Li, J. Wu, Z. Lin, H. Liu, and H. Zha. Recurrent squeeze-and-excitation context aggregation net for single image deraining. *ECCV*, 2018. 1, 2, 5, 6, 8
- [28] Y. Li, R. T. Tan, X. Guo, J. Lu, and M. S. Brown. Rain streak removal using layer priors. In *CVPR*, pages 2736–2744, 2016. 2, 3, 6
- [29] Y. Li, R. T. Tan, X. Guo, J. Lu, and M. S. Brown. Single image rain streak decomposition using layer priors. *IEEE Transactions on Image Processing*, 26(8):3874–3885, 2017. 2
- [30] Y. Luo, Y. Xu, and H. Ji. Removing rain from a single image via discriminative sparse coding. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3397–3405, 2015. 2, 3, 6
- [31] T. Madam Nimisha, K. Sunil, and A. Rajagopalan. Unsupervised class-specific deblurring. In *ECCV*, pages 353–369, 2018. 2
- [32] A. Owens, J. Wu, J. H. McDermott, W. T. Freeman, and A. Torralba. Ambient sound provides supervision for visual learning. In *ECCV*, pages 801–816. Springer, 2016. 3
- [33] N. Ponomarenko, L. Jin, O. Ieremeiev, V. Lukin, K. Egiazarian, J. Astola, B. Vozel, K. Chehdi, M. Carli, F. Battisti, et al. Image database tid2013: Peculiarities, results and perspectives. *Signal Processing: Image Communication*, 30:57–77, 2015. 8

- [34] Z. Ren, J. Yan, B. Ni, B. Liu, X. Yang, and H. Zha. Unsupervised deep learning for optical flow estimation. In *AAAI*, volume 3, page 7, 2017. 2
- [35] S. Roth and M. J. Black. Fields of experts. *International Journal of Computer Vision*, 82(2):205, 2009. 8
- [36] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In *NIPS*, pages 2234–2242, 2016. 2
- [37] R. Stewart and S. Ermon. Label-free supervision of neural networks with physics and domain knowledge. In *AAAI*, volume 1, pages 1–7, 2017. 3
- [38] J. Wang, X. Zhu, S. Gong, and W. Li. Transferable joint attribute-identity deep learning for unsupervised person re-identification. *CVPR*, 2018. 2
- [39] X. Wang and A. Gupta. Unsupervised learning of visual representations using videos. In *ICCV*, pages 2794–2802, 2015. 3
- [40] Y. Wang, S. Liu, C. Chen, and B. Zeng. A hierarchical approach for rain or snow removing in a single color image. *IEEE Transactions on Image Processing*, 26(8):3936–3950, 2017. 4
- [41] Y.-T. Wang, X.-L. Zhao, T.-X. Jiang, L.-J. Deng, Y. Chang, and T.-Z. Huang. Rain streak removal for single image via kernel guided cnn. *arXiv preprint arXiv:1808.08545*, 2018. 3
- [42] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing (TIP)*, 13(4):600–612, 2004. 5
- [43] J. Xu, C. Shi, C. Qi, C. Wang, and B. Xiao. Unsupervised part-based weighting aggregation of deep convolutional features for image retrieval. *AAAI*, 2018. 2
- [44] W. Yang, R. T. Tan, J. Feng, J. Liu, Z. Guo, and S. Yan. Deep joint rain detection and removal from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1357–1366, 2017. 1, 2, 5, 6, 8
- [45] Y. Yang, L. Wen, S. Lyu, and S. Z. Li. Unsupervised learning of multi-level descriptors for person re-identification. In *AAAI*, volume 1, page 2, 2017. 2
- [46] Z. Yi, H. R. Zhang, P. Tan, and M. Gong. Dualgan: Unsupervised dual learning for image-to-image translation. In *ICCV*, pages 2868–2876, 2017. 2
- [47] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. *ICLR*, 2016. 2
- [48] Y. Yuan, S. Liu, J. Zhang, Y. Zhang, C. Dong, and L. Lin. Unsupervised image super-resolution using cycle-in-cycle generative adversarial networks. In *CVPR Workshops*, pages 701–710, 2018. 2
- [49] H. Zhang and V. M. Patel. Convolutional sparse and low-rank coding-based rain streak removal. In *Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on*, pages 1259–1267. IEEE, 2017. 2
- [50] H. Zhang and V. M. Patel. Density-aware single image de-raining using a multi-stream dense network. *CVPR*, 2018. 1, 2, 3, 5, 6, 8
- [51] J. Zhang, T. Zhang, Y. Dai, M. Harandi, and R. Hartley. Deep unsupervised saliency detection: A multiple noisy labeling perspective. In *CVPR*, pages 9029–9038, 2018. 2
- [52] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017. 8
- [53] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *ICCV*, 2017. 1, 2, 4, 5
- [54] L. Zhu, C.-W. Fu, D. Lischinski, and P.-A. Heng. Joint bi-layer optimization for single-image rain streak removal. In *ICCV*, pages 2526–2534, 2017. 2, 3